

IFI TECHNICAL REPORTS

Institute of Computer Science,
Clausthal University of Technology

IfI-05-05

Clausthal-Zellerfeld 2005

Barycentric Coordinates for Arbitrary Polygons in the Plane

Kai Hormann

Institute of Computer Science, Clausthal University of Technology, Germany

Abstract

Barycentric coordinates for triangles are commonly used in computer graphics, geometric modelling, and other computational sciences for various purposes. In this paper, we extend this well-known concept and propose a generalization to sets of arbitrary polygons in the plane. Besides many other important properties, our coordinate functions are local and smooth and allow an efficient and robust implementation. These coordinates are particularly useful for interpolating data that is given at the vertices of the polygons and we present several examples of their application to common problems in computer graphics and geometric modelling.

1 Introduction

It follows from *Ceva's Theorem*¹ (Ceva, 1678) that for any point v inside a planar triangle $[v_1, v_2, v_3]$ there exist three masses w_1 , w_2 , and w_3 , such that, if placed at the corresponding vertices of the triangle, their centre of mass (or barycentre²) will coincide with v , i.e.,

$$\frac{w_1 v_1 + w_2 v_2 + w_3 v_3}{w_1 + w_2 + w_3} = v. \quad (1)$$

Möbius³ was the first to study such *mass points* and he defined w_1 , w_2 , and w_3 as the *barycentric coordinates* of v (Möbius, 1827). Evidently, these barycentric coordinates are only unique up to multiplication by a common non-zero scalar and they are usually *normalized* to sum to one.

These normalized triangular barycentric coordinates are linear in v and have the additional property that the i -th coordinate has value 1 at v_i and 0 at the other v_j . This is why they are commonly used to linearly interpolate values given at the vertices of a triangle and have applications in computer graphics (e.g. Gouraud and Phong shading, texture mapping, ray-triangle-intersection), geometric modelling (e.g. triangular Bézier patches, splines over triangulations), and many other fields (e.g. the finite element method, terrain modelling).

¹Giovanni Ceva, 1647–1734

²from greek: βαρύς (barys) = heavy, deep, dense

³August Ferdinand Möbius, 1790–1868

In many applications it would be useful to have a generalization of barycentric coordinates to arbitrary n -sided polygons or even sets of polygons in the plane with vertices v_1, \dots, v_n . One would then like to have smooth *homogeneous barycentric coordinates* $w_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ that generalize Equation (1),

$$\sum_{i=1}^n w_i(v)(v_i - v) = 0, \quad (2)$$

and associated *normalized barycentric coordinates*,

$$\lambda_i(v) = \frac{w_i(v)}{\sum_{j=1}^n w_j(v)}, \quad (3)$$

so that any point v in the plane can be written as an *affine combination* of v_1, \dots, v_n with weights $\lambda_1(v), \dots, \lambda_n(v)$. Furthermore, these coordinates should satisfy the *Lagrange property*

$$\lambda_i(v_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \quad (4)$$

As we will show in Section 3 there are many ways of defining homogeneous barycentric coordinates, but for most choices the normalized coordinates in (3) either are not well-defined everywhere in \mathbb{R}^2 , or do not meet the constraints in (4). Nevertheless, we present a particular choice that fulfills all properties. Our generalized barycentric coordinates have a number of other important properties and enable a very efficient and robust implementation as shown in Section 4.

The main application of these coordinates is interpolation and in Section 5 we show several examples from computer graphics and geometric modelling that can be seen as interpolation problems and hence can be solved with our approach. In particular, we propose an improved Phong shading method for non-triangular faces, a simple image warping technique, and interpolation of data that is specified on planar curves.

2 Related Work

2.1 Barycentric Coordinates

Most of the previous work on barycentric coordinates discusses the extension to *convex* polygons. The first such generalization appears in the pioneering work of Wachspress (1975) who was interested in extending the finite element method. These *Wachspress coordinates* are rational polynomials and were later generalized to convex polytopes by Warren (1996) who also showed that they have minimal degree (Warren, 2003). They can be computed with simple and local formulas in the plane (Meyer et al., 2002) as well as in higher dimensions (Warren et al., 2003) and have many other nice properties like affine invariance. An extension of Wachspress coordinates to *weakly convex* polygons

has been suggested by Malsch and Dasgupta (2004b) but their construction is iterative and the resulting coordinate functions cannot be expressed in a local form.

Other generalizations of barycentric coordinates to convex polygons and even to the kernel of a star-shaped polygon were presented in the context of triangle mesh parameterization, for example the *discrete harmonic* (Pinkall and Polthier, 1993; Eck et al., 1995) and the *mean value coordinates* (Floater, 2003). Also the natural neighbour interpolants that were proposed by Sibson (1980, 1981) for the purpose of scattered data interpolation provide barycentric coordinates for convex polygons, but like the coordinates in (Farin, 1990) they are not more than C^1 -continuous away from the data points. Hiyoshi and Sugihara (2000) have recently extended Sibson's approach and presented C^k -continuous coordinates, but their computation is very costly and involves numerical integration.

Except for the discrete harmonic coordinates, all these coordinates have in common that they are *positive* over the interior of any convex polygon. In fact, this property has often been used in the definition of barycentric coordinates instead of the weaker interpolation condition (4) which is in any case a consequence of positivity in the convex case, as discussed by Floater et al. (2005). They also proved that for a convex polygon the Wachspress and the mean value coordinates are the only positive coordinates with uniform scaling invariance that can be computed with a local three-point-formula.

For non-convex polygons a usual approach is to triangulate the domain and apply the standard barycentric coordinates on each triangle, but the result depends on the particular triangulation chosen and is only C^0 -continuous over the edges of the triangles. To the best of our knowledge, the recent paper by Malsch and Dasgupta (2004a) is the only one that addresses the construction of smooth coordinates for non-convex polygons. Their coordinate functions are well-defined over the convex hull of any concave polygon with possible holes.

Contributions. The barycentric coordinates that we present are more general as they can be constructed for any set of non-intersecting polygons and are well-defined everywhere in the plane. They are smooth (i.e. C^∞) except at the vertices of the polygons where they are only C^0 and can be computed with a simple and local formula. These properties make them an ideal tool for the interpolation of data that is given at the vertices.

2.2 Interpolation

The interpolation of data that is given at the vertices of a set of polygons can be seen as a scattered data interpolation (SDI) problem and many different approaches exist to solve it, including radial basis functions (Beatson et al., 1999; Buhmann, 2000) and bivariate splines (Lee et al., 1997; Nürnberger and Zeilfelder, 2000). Such interpolation problems frequently occur in various fields of science and engineering (e.g. geology, reverse engineering, numerical simulation) but also in computer graphics and geometric modelling.

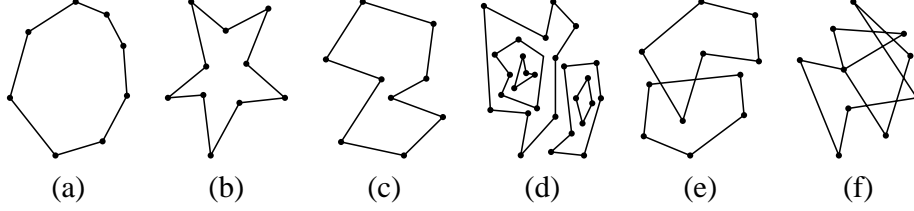


Figure 1: We consider convex (a), star-shaped (b), simple (c), and sets of simple polygons (d), but not polygons with intersecting edges (e) or multiple vertices (f).

One example is *image warping* (see Wolberg (1990); Glasbey and Mardia (1998); Milliron et al. (2002) for an overview of the state-of-the-art) and radial basis functions (Arad et al., 1994; Ruprecht and Müller, 1995) as well as B-splines (Lee et al., 1995, 1997) have been used in this context.

Another important problem is that of *transfinite interpolation* where the data to be interpolated is given as functions over a set or network of planar curves. There exist a number of well-established methods for some special cases, like Coons' or Gordon surfaces (Farin, 2002) for triangular- or rectangular-shaped input curves, but very few are known for the general case. The standard approach is to either sample the data and apply an SDI method or to solve a partial differential equation (PDE) with the given data as boundary conditions (Chai et al., 1998; Kounchev, 2001). But like the generalization of Sibson's interpolants that was suggested by Gross and Farin (1999), this is usually very costly to compute.

Contributions. Due to the Lagrange property of our generalized barycentric coordinates, interpolation of data that is given at the vertices of a set of polygons can be done directly and efficiently without solving a linear system. In the context of transfinite interpolation, the locality of our coordinate functions further enables a simple and progressive update of the solution if the sampling density is increased. Interestingly, our interpolating surfaces are often strikingly similar to interpolating thin plate splines, even though our approach requires far less computational effort.

3 Definition and Properties

Let Ψ be an arbitrary polygon or a set of arbitrary polygons in the plane with $n \geq 3$ distinct vertices v_1, \dots, v_n and non-intersecting (open) edges $e_i = (v_i, v_{i+1}) = \{(1 - \mu)v_i + \mu v_{i+1} : 0 < \mu < 1\}$; see Figure 1 for some examples. If Ψ is a set of (possibly nested) polygons, we require the orientations of the polygons to alternate as shown in Figure 14 (a) in Appendix B.

We define for any $v \in \mathbb{R}^2$ the usual Euclidian distance $r_i(v) = \|v_i - v\|$ to v_i and denote by $\alpha_i(v)$ the *signed* angle in the triangle $[v, v_i, v_{i+1}]$ at the vertex v . Then

$$A_i(v) = r_i(v)r_{i+1}(v) \sin(\alpha_i(v))/2 \quad (5)$$

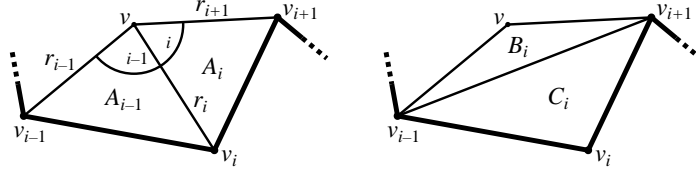


Figure 2: Notation used for angles, areas, and distances.

and

$$B_i(v) = r_{i-1}(v)r_{i+1}(v) \sin(\alpha_{i-1}(v) + \alpha_i(v))/2 \quad (6)$$

are the *signed* areas of the triangles $[v, v_i, v_{i+1}]$ and $[v, v_{i-1}, v_{i+1}]$, respectively⁴; see Figure 2. It is well-known (Coxeter, 1969) that

$$A_i(v), \quad -B_i(v), \quad A_{i-1}(v)$$

are the homogeneous barycentric coordinates for any $v \in \mathbb{R}^2$ with respect to the triangle $\triangle_i = [v_{i-1}, v_i, v_{i+1}]$, in other words,

$$A_i(v)(v_{i-1} - v) - B_i(v)(v_i - v) + A_{i-1}(v)(v_{i+1} - v) = 0. \quad (7)$$

Note that every vertex v_i of Ψ has a corresponding coordinate in each of the three triangles \triangle_{i-1} , \triangle_i , and \triangle_{i+1} . We can now take for every v_i a weighted average of these three coordinates and define

$$w_i(v) = b_{i-1}(v)A_{i-2}(v) - b_i(v)B_i(v) + b_{i+1}(v)A_{i+1}(v), \quad (8)$$

where the weight functions $b_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ can be chosen arbitrarily. Then it follows immediately from (7) that these functions w_i are homogeneous barycentric coordinates with respect to Ψ , i.e. they satisfy Equation (2).

The critical part now is the normalization of these homogeneous coordinates, i.e. to guarantee that the denominator in (3) is non-zero for every $v \in \mathbb{R}^2$.

For convex polygons this is relatively easy to achieve. Indeed, it can be derived from (8) that

$$W(v) = \sum_{i=1}^n w_i(v) = \sum_{i=1}^n b_i(v)C_i, \quad (9)$$

where $C_i = A_{i-1}(v) + A_i(v) - B_i(v)$ is the signed area of \triangle_i ; see Figure 2. Now, if Ψ is convex then all C_i have the same sign (which depends on the orientation of Ψ) and so as long as all the weight functions b_i are positive (or negative) then $W(v)$ can never be zero.

⁴Note that we always treat indices cyclically with respect to each of the components in Ψ . Hence, if Ψ is a set of two polygons with n_1 and n_2 vertices, then $e_{n_1} = (v_{n_1}, v_1)$ and $B_{n_1+1}(v)$ is the area of $[v, v_{n_1+n_2}, v_{n_1+2}]$.

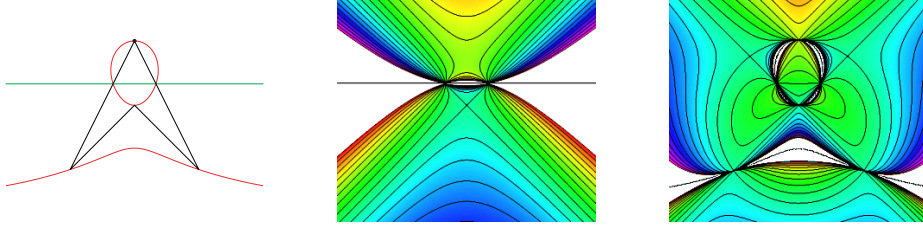


Figure 3: Zero-set of the denominator $W(v)$ for the Wachspress (green) and the discrete harmonic coordinates (red) for a concave polygon (black) and contour plots of the normalized coordinate function λ_i that corresponds to the topmost vertex.

But in the general case that we consider it is more difficult to avoid dividing by zero. Consider, for example, the *Wachspress* and *discrete harmonic coordinates* that can be generated by the weight functions

$$b_i^W(v) = \frac{1}{A_{i-1}(v)A_i(v)} \quad \text{and} \quad b_i^D(v) = \frac{r_i(v)^2}{A_{i-1}(v)A_i(v)}.$$

Both coordinates are well-defined for v inside any convex polygon. However, inside a non-convex polygon, $W(v)$ can become zero, and the normalized coordinates $\lambda_i(v)$ may have non-removable poles, as illustrated for a concave quadrilateral in Figure 3.

To avoid this problem, Equation (9) suggests taking weight functions like $b_i(v) = 1/C_i$ so that $W(v) = n$. But although this particular choice gives well-defined (and linear) normalized coordinates λ_i as long as no three consecutive vertices of Ψ are collinear, they unfortunately do not satisfy Equation (4).

However, we found that the weight functions

$$b_i(v) = \frac{r_i(v)}{A_{i-1}(v)A_i(v)}$$

guarantee $W(v) \neq 0$ for any $v \in \mathbb{R}^2$ (see Appendix B), and at the same time, the corresponding normalized coordinates have the Lagrange property (see Appendix C). For this particular choice of b_i , Equation (8) becomes

$$w_i(v) = \frac{r_{i-1}(v)A_i(v) - r_i(v)B_i(v) + r_{i+1}(v)A_{i-1}(v)}{A_{i-1}(v)A_i(v)}, \quad (10)$$

which we recognize as the *mean value coordinates* in the form given by Floater et al. (2005). By using (5) and (6) and some trigonometric identities this formula simplifies to

$$\frac{w_i(v)}{2} = \frac{\tan(\alpha_{i-1}(v)/2) + \tan(\alpha_i(v)/2)}{r_i(v)}, \quad (11)$$

which is the formula that originally appeared in (Floater, 2003).

Let us now summarize the properties of the normalized barycentric coordinates λ_i that are defined by these w_i and Equation (3):

1. **Affine precision.** It follows from (2) and (3) that

$$\sum_{i=1}^n \lambda_i(v) \varphi(v_i) = \varphi(v)$$

for any affine function $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ and $v \in \mathbb{R}^2$.

2. **Partition of unity.** With $\varphi(v) = 1$ it follows immediately from the previous property that $\sum_{i=1}^n \lambda_i(v) = 1$ for any $v \in \mathbb{R}^2$.
3. **Lagrange property.** The functions λ_i satisfy $\lambda_i(v_j) = \delta_{i,j}$ (see Appendix C).
4. **Linearity property.** The functions λ_i are linear along the edges e_j of Ψ (see Appendix C).
5. **Linear independence.** An immediate consequence of the previous property is that if $\sum_{i=1}^n c_i \lambda_i(v) = 0$ for all $v \in \mathbb{R}^2$ then all c_i must be zero.
6. **Smoothness.** The functions λ_i are C^∞ everywhere except at the vertices v_j where they are only C^0 (see Appendix C).
7. **Similarity invariance.** Since the homogeneous coordinates w_i depend only on areas and distances and since any uniform scale factor cancels out in the normalization (3), it follows that if $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a similarity transformation⁵ and $\hat{\lambda}_i$ are the normalized barycentric coordinates with respect to $\hat{\Psi} = \varphi(\Psi)$, then $\lambda_i(v) = \hat{\lambda}_i(\varphi(v))$.
8. **Locality.** The homogeneous coordinate function w_i depends only on the vertices v_{i-1} , v_i , and v_{i+1} of Ψ .
9. **Refinability.** If we refine Ψ to $\hat{\Psi}$ by adding the vertex $\hat{v} = (1 - \mu)v_j + \mu v_{j+1}$, and denote by $\hat{\lambda}_i$ and $\hat{\lambda}$ the normalized barycentric coordinates with respect to $\hat{\Psi}$ then we have $\lambda_j = \hat{\lambda}_j + (1 - \mu)\hat{\lambda}$, $\lambda_{j+1} = \hat{\lambda}_{j+1} + \mu\hat{\lambda}$, and $\lambda_i = \hat{\lambda}_i$ for $i \neq j, j+1$ (see Appendix A).
10. **Positivity.** It follows from (11) that the functions λ_i are positive inside the kernel of a star-shaped polygon and in particular inside a convex polygon.

Figure 4 illustrates the typical behaviour of the functions λ_i for a set of simple polygons.

The main application of these generalized barycentric coordinates is the interpolation of values that are given at the vertices v_i of Ψ . In other words, if a data value $f_i \in \mathbb{R}^d$ is specified at each v_i , then we are interested in the function $F : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ that is defined by

$$F(v) = \sum_{i=1}^n \lambda_i(v) f_i = \sum_{i=1}^n w_i(v) f_i / W(v). \quad (12)$$

⁵a translation, rotation, uniform scaling, or combination of these

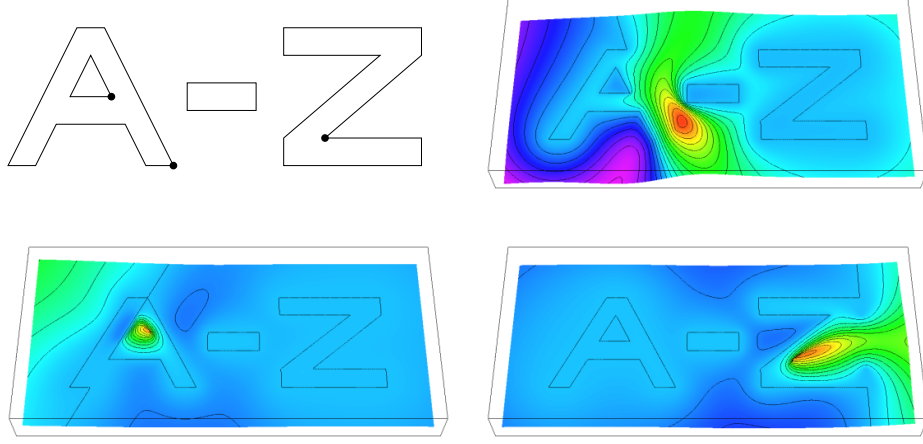


Figure 4: In this example, Ψ is a set of four simple polygons and we show contour plots of three of our generalized barycentric coordinate functions λ_i . They equal 1 at the vertex v_i , equal 0 at all other vertices of Ψ , and are linear along edges.

Due to the Lagrange and linearity properties of the coordinates, this function interpolates f_i at v_i and is linear along the edges e_i of Ψ . An example where the f_i are colour values is shown in Figure 5.

4 Implementation

For the actual computation of the interpolation function F we use Equation (11) to determine the homogeneous coordinates $w_i(v)$ and their sum $W(v)$, but we suggest a slight modification that avoids the computation of the angles $\alpha_i(v)$ and enables an efficient handling of the special cases that can occur. If we let $s_i(v) = v_i - v$ and denote the dot product of $s_i(v)$ and $s_{i+1}(v)$ by $D_i(v)$, then we have

$$\tan(\alpha_i(v)/2) = \frac{1 - \cos(\alpha_i(v))}{\sin(\alpha_i(v))} = \frac{r_i(v)r_{i+1}(v) - D_i(v)}{2A_i(v)},$$

and we use this formula as long as $A_i(v) \neq 0$. Otherwise, v lies on the line through v_i and v_{i+1} and we distinguish three cases. If $v = v_i$ or $v = v_{i+1}$, then we do not bother to compute the $w_i(v)$ and simply set $F(v) = f_i$ or $F(v) = f_{i+1}$. Likewise, if v lies on the edge e_i , then we use the linearity of F along e_i to determine $F(v)$ directly. Note that we can easily identify this case because it implies $D_i(v) < 0$. Finally, if v is not on Ψ then we conclude that $\alpha_i(v) = 0$ and therefore $\tan(\alpha_i(v)/2) = 0$.

The pseudo-code for computing $F(v)$ is given in Figure 6. Note that we can also use this approach to compute each single coordinate function λ_j by simply setting $f_i = \delta_{ij}$.

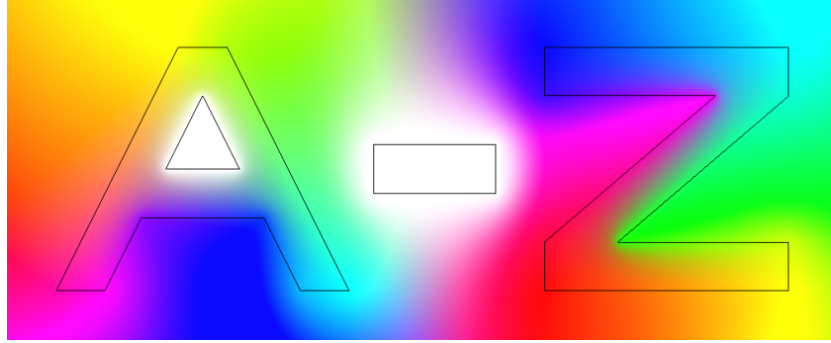


Figure 5: Colour interpolation with generalized barycentric coordinates for the polygons from Figure 4.

```

function  $F(v)$ 
  for  $i = 1$  to  $n$  do
     $s_i := v_i - v$ 
  for  $i = 1$  to  $n$  do
     $r_i := \|s_i\|$ 
     $A_i := \det(s_i, s_{i+1})/2$ 
     $D_i := \langle s_i, s_{i+1} \rangle$ 
    if  $r_i = 0$  then                                     //  $v = v_i$ 
      return  $f_i$ 
    if  $A_i = 0$  and  $D_i < 0$  then                         //  $v \in e_i$ 
       $r_{i+1} = \|s_{i+1}\|$ 
      return  $(r_{i+1}f_i + r_i f_{i+1})/(r_i + r_{i+1})$ 
   $f := 0$ 
   $W := 0$ 
  for  $i = 1$  to  $n$  do
     $w := 0$ 
    if  $A_{i-1} \neq 0$  then
       $w := w + (r_{i-1} - D_{i-1}/r_i)/A_{i-1}$ 
    if  $A_i \neq 0$  then
       $w := w + (r_{i+1} - D_i/r_i)/A_i$ 
     $f := f + w f_i$ 
     $W := W + w$ 
  return  $f/W$ 

```

Figure 6: Pseudo-code for evaluating the interpolation function.

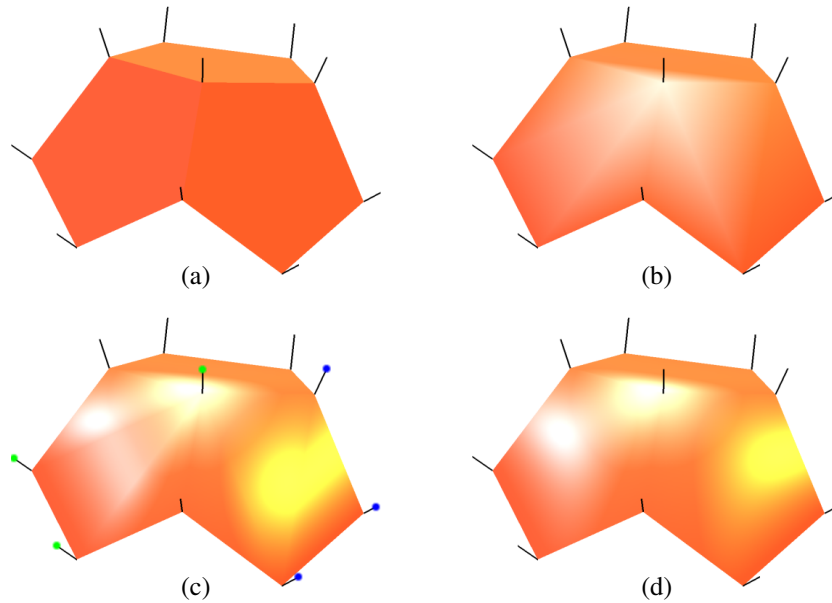


Figure 7: Rendering of three pentagons with flat shading (a), Gouraud shading (b), Phong shading (c), and generalized Phong shading (d).

5 Applications and Results

We now present three applications of our generalized barycentric coordinates that demonstrate their potential impact on computer graphics and geometric modelling.

5.1 Phong Shading for Arbitrary Polygons

The standard approach to treating an arbitrary polygon in the rendering pipeline is to first tessellate it into triangles and then process each triangle in turn. For example, *OpenGL* automatically splits a pentagonal face into three triangles. This gives the expected result if flat shading is used, but the splits become visible as soon as Gouraud shading or Phong shading is turned on; see Figure 7 (a–c).

Phong shading uses triangular barycentric coordinates to linearly interpolate the normals that are given at the vertices of the face over each generated triangle. This can be seen for the two triangles that are marked by the green and blue spots in the Phong-shaded result (c). Since the lower two vertices of both triangles have identical normals, the interpolated normal as well as the resulting colour value vary linearly inside the triangles. Instead, we can improve the idea of Phong shading and use the generalized barycentric coordinates to smoothly interpolate the normals over the whole polygon, giving a much more pleasant rendering result; see Figure 7 (d).

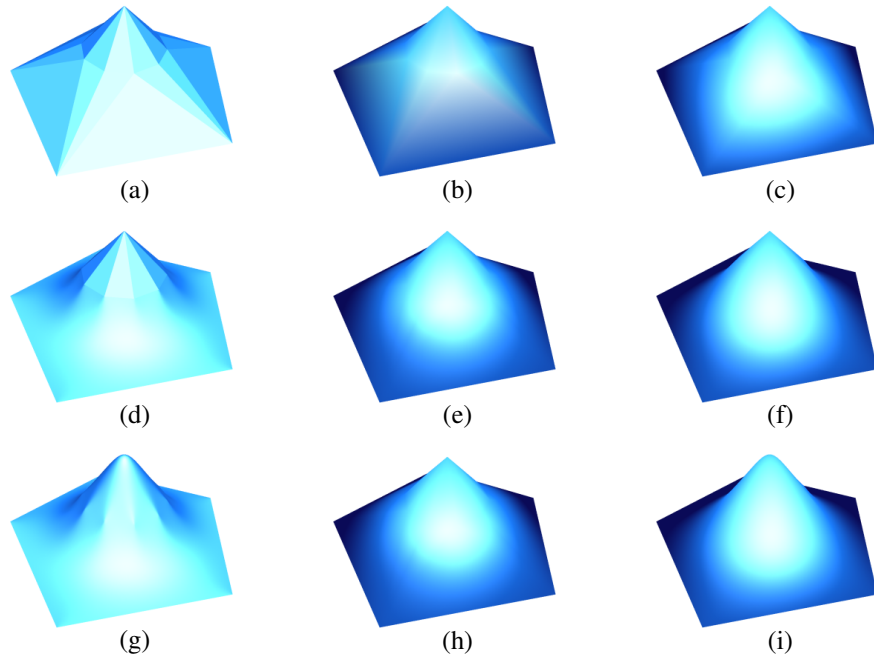
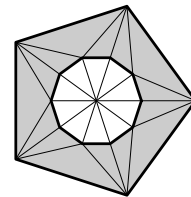


Figure 8: Flat (a), Gouraud (b), and Phong shading (c) of a simple triangle mesh. Generalized barycentric coordinates can be used to smoothly interpolate the vertices of the exterior ring of triangles (d) and their normals. The interpolated normals can then be used to shade the triangle mesh (e) or the interpolated geometry (f). This approach can be extended to the interior triangle fan (g–i).

Another example is shown in Figure 8. We first took the flat configuration on the right as a parameterization of the triangle mesh in (a) and regarded the two rings of vertices as polygons (thick lines). Then we smoothly interpolated the normals that were specified at these vertices over the exterior ring of triangles (grey area) with generalized barycentric coordinates and used the result for a superior shading of the triangle mesh (e).



The generalized barycentric coordinates further allowed us to interpolate the vertex positions instead of the normals and to create the smooth blend surface between the two polygons that is shown in (d) with shading according to the surface normal and in (f) with shading according to the interpolated normals. Of course, the interpolation of normals and geometry is not limited to the grey region and can be extended to the whole triangle mesh (g–i). Note that the interpolation surface in (g) is defined by all vertices of the triangle mesh except for the central vertex and therefore does not necessarily pass through it.

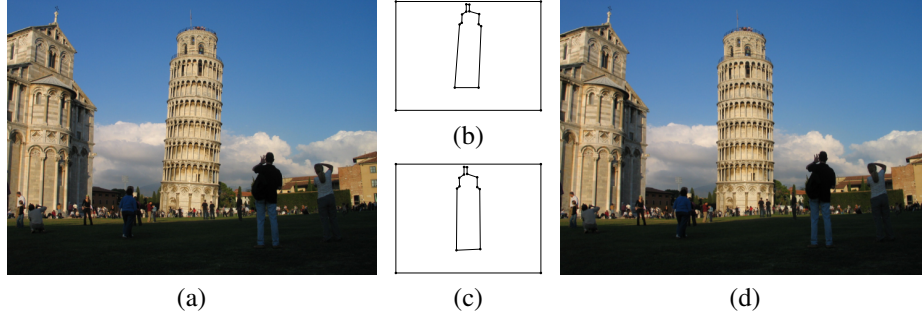


Figure 9: The Leaning Tower of Pisa (a) can be straightened by our barycentric warp function which maps the two source polygons (b) to the target polygons (c). The Straight Tower of Pisa is shown in (d).

5.2 Image Warping

Another potential application of the generalized barycentric coordinates is image warping as they offer a particularly simple solution to this problem that can briefly be stated as follows.

Given a rectangular region Ω , a set of *source polygons* Ψ with vertices $v_i \in \Omega$, and a topologically equivalent⁶ set of *target polygons* $\hat{\Psi}$ with vertices $\hat{v}_i \in \Omega$, we would like to construct a smooth *warp function* $f : \Omega \rightarrow \Omega$ that maps each v_i to \hat{v}_i . This warp function can then be used to deform a *source image* $I : \Omega \rightarrow C$ that maps Ω to some colour space C into a *target image* $\hat{I} : \Omega \rightarrow C$ by simply setting $\hat{I} = I \circ f^{-1}$. For practical reasons, the *inverse mapping* $g = f^{-1}$ is often constructed instead of f .

Such an inverse warp function can easily be defined with the generalized barycentric coordinates $\hat{\lambda}_i$ of $\hat{\Psi}$. It follows immediately from the Lagrange property that the function $g : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with

$$g(x) = \sum_{i=1}^n \hat{\lambda}_i(x) v_i \quad (13)$$

maps each \hat{v}_i to v_i and thus defines a proper inverse warp function. The warped image can now be generated by simply setting the colour of each target pixel x in \hat{I} to the colour of the source point $g(x)$ in I . In our examples we used a simple bilinear interpolation to determine $I(g(x))$ from the 2×2 grid of pixels surrounding $g(x)$.

Like warping with B-splines and radial basis functions with linear precision, this *barycentric warp* reproduces affine transformations. In other words, if $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is an affine transformation and $\hat{\Psi} = \varphi(\Psi)$ then $\hat{I} = I \circ \varphi^{-1}$. Indeed, since $v_i = \varphi^{-1}(\hat{v}_i)$ it follows from the affine precision property of the barycentric coordinates that $g(x) = \varphi^{-1}(x)$ in (13).

⁶with the same number of vertices and components

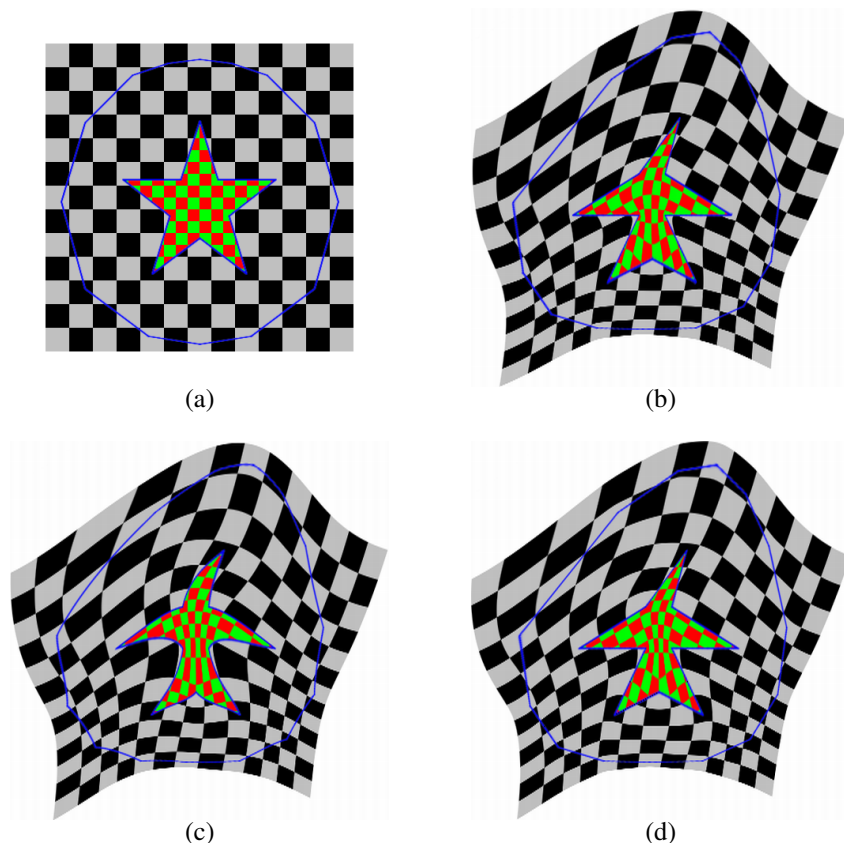


Figure 10: Warping an image (a) with generalized barycentric coordinates (b) and radial basis functions (c,d) by moving the vertices of two nested polygons (blue).

Another property of the barycentric warp is that it is linear along the edges of the polygons. Figure 10 (b) shows the result of warping the image in (a) after moving the vertices of two nested polygons. The exterior polygon with 14 vertices controls the global shape of the warp, while the interior polygon with 10 vertices is used to deform the star. The result is smooth and the star is clearly mapped to a star with straight edges.

For comparison, we also generated the *thin plate spline* warp (c) using radial basis interpolation with basis function $\phi(r) = r^2 \log r$ (see Arad et al. (1994) for details). This warp does not reproduce the straight edges of the star, but a common trick to overcome this drawback is to sample the edges with additional vertices. For example, by taking 20 samples per edge, we obtained the result in (d). It is very similar to the barycentric warp, but takes considerably longer to compute: 20 seconds for a 600×600 image on a 2.8 Ghz Pentium, while the barycentric warp took less than one second.

This is due to the large linear system that needs to be solved and the large number of basis functions that have to be evaluated. In general, if n is the number of vertices in Ψ and k is the number of samples per edge, then solving the linear system with a standard method is an $O(k^3 \cdot n^3)$ operation and evaluating the m pixels of \hat{I} is of order $O(m \cdot k \cdot n)$. Of course there exist specialized solvers that can reduce the cost for solving the linear system, but the advantage of the barycentric warp is that it does not require any system to be solved and the whole computation is of order $O(m \cdot n)$ only.

We conclude that the barycentric warp is particularly useful whenever straight edges need to be preserved. For example, making the boundary of the rectangular image one of the source and the target polygons guarantees that the warped image will be rectangular too; see Figure 9.

5.3 Transfinite Interpolation

Our generalized barycentric coordinates also provide an efficient solution to the following interpolation problem. Given a set of closed planar curves c_j and some d -dimensional data over these curves $d_j : c_j \rightarrow \mathbb{R}^d$ we would like to have a function $F : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ that interpolates the given data, i.e. $F(v) = d_j(v)$ for any $v \in c_j$.

The obvious approach is to approximate the given curves c_j with a set of polygons Ψ whose vertices v_i lie on the curves c_j . Then the function F in Equation (12) with $f_i = d_j(v_i)$ clearly is an approximate solution of the stated interpolation problem. And as the polygons Ψ converge to the curves c_j by increasing the sampling density, so does the function F converge to the desired solution.

The interesting fact now is that the structure of our generalized barycentric coordinates allows an efficient update of the solution if the sampling density is increased. Assume that we computed $F(v)$ and also $W(v)$ from Equation (9) for some polygon Ψ and then refine Ψ to $\hat{\Psi}$ by adding a vertex \hat{v} with data value $\hat{f} = d_j(\hat{v})$ between v_j and v_{j+1} . Then it follows from Appendix A that the refined interpolation function $\hat{F}(v)$ can be written as

$$\hat{F}(v) = F(v) + \frac{\hat{w}(v)}{\widehat{W}(v)}(\hat{f} - \rho(v)F(v) - \sigma(v)f_j - \tau(v)f_{j+1}) \quad (14)$$

with

$$\widehat{W}(v) = W(v) + \rho(v)\hat{w}(v) \quad (15)$$

where $\hat{w}(v)$ is the new homogeneous coordinate function for \hat{v} and $\rho(v)$, $\sigma(v)$, and $\tau(v)$ are the normalized barycentric coordinates of \hat{v} with respect to the triangle $[v, v_j, v_{j+1}]$. Note that ρ , σ , and τ can be computed from the same ingredients as \hat{w} .

In the example in Figure 11, we exploited this recurrence relation as follows. We first sampled the three given curves uniformly with 100 vertices v_i and computed the curve normals n_i at v_i . Then we evaluated the interpolating function F as well as W on a regular 512×512 grid, yielding values F_{kl} and W_{kl} with $0 \leq k, l \leq 511$. We

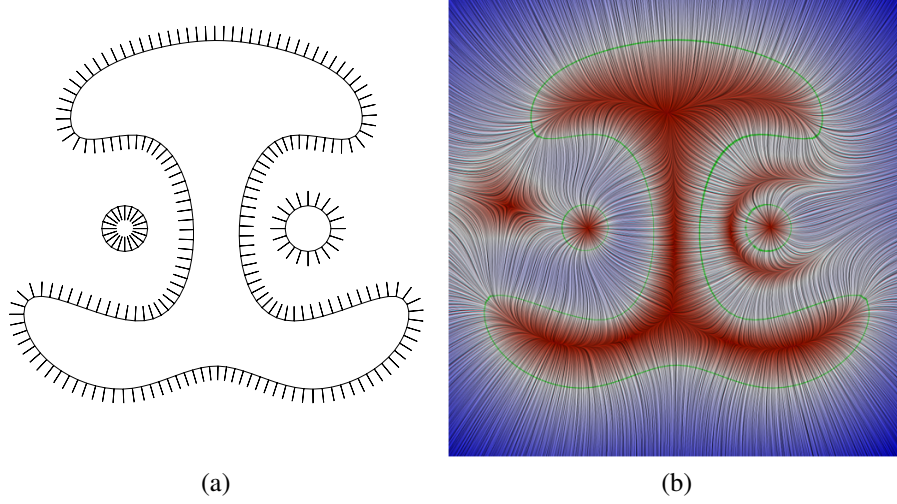


Figure 11: Three curves with normal vectors (a) and a LIC visualization of the interpolated vector field (b). The colour refers to the length ℓ of the vectors (white: $\ell = 1$, blue: $\ell > 1$, red: $\ell < 1$).

finally refined Ψ successively by splitting the edge with the largest approximation error to the curve and updated F_{kl} and W_{kl} according to (14) and (15). We stopped the refinement process as soon as $\max_{k,l} \|\hat{F}_{kl} - F_{kl}\| < 0.001$, i.e. when the maximum update went below 1% of the length of the given vectors. The final polygon consisted of 357 vertices and it took about 10 seconds to compute the final values F_{kl} .

Another example where we interpolated height data is shown in Figure 12. We first used Shewchuk's *triangle* program (Shewchuk, 2002) to generate a regular triangulation of the domain and the 2,500 vertices and edges of Ψ . Then we evaluated the interpolating function at the 100,000 vertices of this constrained Delaunay triangulation, which took about 25 seconds. We also computed the interpolating thin plate spline for comparison and found the height difference between both surfaces to be less than 10% of the maximum value.

6 Conclusions

We have presented a natural generalization of triangular barycentric coordinates both to arbitrary polygons and even to sets of polygons. Our generalized barycentric coordinates have a number of important properties and are particularly useful for the interpolation of data that is given at the vertices of the polygons. As shown in Figures 10 and 12, the behaviour of the interpolant is strikingly similar to that of thin plate splines. However, unlike thin plate splines, our interpolant is extremely fast to evaluate. In our

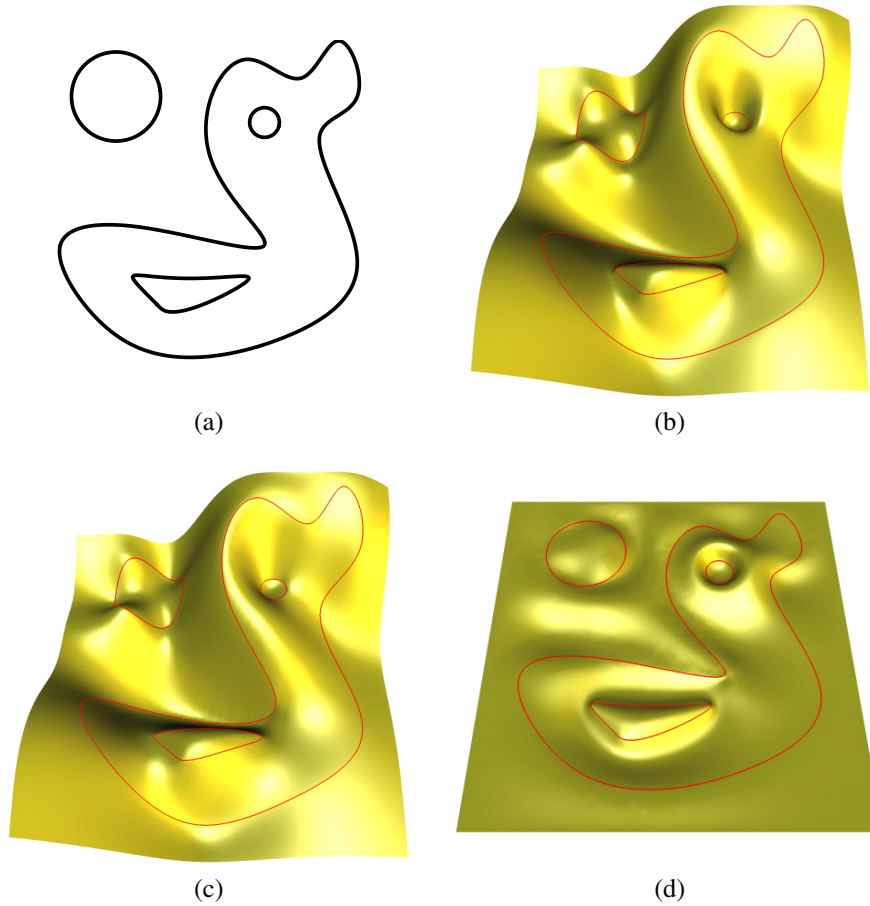


Figure 12: Interpolation of height values (red) that are given as smooth functions over the four curves in (a) with generalized barycentric coordinates (b) and with a thin plate spline (c). The difference of both surfaces is shown in (d).

implementation we have typically seen around 10,000,000 evaluations of the coordinates λ_i per second.

We note that, in contrast to Sibson's coordinates for scattered data, our coordinates have global support and are not everywhere positive in the case of arbitrary polygons. On the other hand it is probably due to these properties that our interpolants behave so nicely. In the context of image warping, this lack of positivity means that our barycentric warp function is not guaranteed to be one-to-one, except if both the source and the target polygon are convex. But nevertheless the method appears to work very well in practice.

7 Future Work

An interesting question for future research is whether our generalized barycentric coordinates can be implemented in the graphics hardware to further speed up the computation. This would be useful for image warping and even more for our improved Phong shading. The algorithm for evaluating the interpolation function seems to be simple enough to be coded in the fragment shader of the GPU and could further be simplified because the entities $s_i(v)$ and $A_i(v)$ are linear functions and destined to be generated by the rasterizer through linear interpolation of the values computed at the vertices v_i . However, as long as the graphics hardware does not provide loops and the code must therefore be replicated n times, and as long as the code length of programs on the GPU is restricted, the implementation will be limited to work only for small n .

We also plan to investigate two very promising extensions of our coordinates. Firstly, we are confident that they can be generalized to polyhedra of arbitrary genus in 3D, leading to an analogous transfinite interpolation of data over closed surfaces. Secondly, we believe that our barycentric coordinates can be generalized from polygons to smooth closed curves or sets of curves in the plane. We will thoroughly analyse the mathematical and computational aspects of both of these generalizations.

Finally, we would like to be able to interpolate data over curves, where not only height data are specified but also tangent data, and possibly higher order derivatives. Our coordinates do not currently allow such Hermite interpolation, but we have some ideas on how this might be achieved, and we will explore these in future work.

References

- Arad, N., Dyn, N., Reisfeld, D., and Yeshurun, Y. (1994). Image warping by radial basis functions: Application to facial expressions. *CVGIP: Graphical Models and Image Processing*, 56(2):161–172.
- Beatson, R. K., Cherrie, J. B., and Mouat, C. T. (1999). Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Advances in Computational Mathematics*, 11(2-3):253–270.
- Buhmann, M. (2000). Radial basis functions: the state-of-the-art and new results. *Acta Numerica*, 9:1–37.
- Ceva, G. (1678). *De lineis rectis se invicem secantibus, statica constructio*. Ludovici Montiae, Mediolanum.
- Chai, J., Miyoshi, T., and Nakamae, E. (1998). Contour interpolation and surface reconstruction of smooth terrain models. In *Proceedings of the Conference on Visualization '98*, pages 27–33.

- Coxeter, H. S. M. (1969). *Introduction to Geometry*, chapter 13.7, Barycentric coordinates, pages 216–221. Wiley, New York, second edition.
- Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. (1995). Multiresolution analysis of arbitrary meshes. In *Proceedings of SIGGRAPH '95*, pages 173–182.
- Farin, G. (1990). Surfaces over Dirichlet tessellations. *Computer Aided Geometric Design*, 7(1-4):281–292.
- Farin, G. (2002). *Curves and Surfaces for CAGD: A Practical Guide*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, San Francisco, fifth edition.
- Floater, M. S. (2003). Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27.
- Floater, M. S., Hormann, K., and Kós, G. (2005). A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*. To appear.
- Glasbey, C. A. and Mardia, K. V. (1998). A review of image warping methods. *Journal of Applied Statistics*, 25(2):155–171.
- Gross, L. and Farin, G. (1999). A transfinite form of Sibson’s interpolant. *Discrete Applied Mathematics*, 93(1):33–50.
- Hiyoshi, H. and Sugihara, K. (2000). Voronoi-based interpolation with higher continuity. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 242–250.
- Kounchev, O. (2001). *Multivariate Polysplines: Applications to Numerical and Wavelet Analysis*. Academic Press, London, San Diego.
- Lee, S., Chwa, K.-Y., Shin, S. Y., and Wolberg, G. (1995). Image metamorphosis using snakes and free-form deformations. In *Proceedings of SIGGRAPH '95*, pages 439–448.
- Lee, S., Wolberg, G., and Shin, S. Y. (1997). Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244.
- Malsch, E. A. and Dasgupta, G. (2004a). Algebraic construction of smooth interpolants on polygonal domains. *The Mathematica Journal*, 9(3).
- Malsch, E. A. and Dasgupta, G. (2004b). Interpolations for temperature distributions: A method for all non-concave polygons. *International Journal of Solids and Structures*, 41(8):2165–2188.

- Meyer, M., Lee, H., Barr, A. H., and Desbrun, M. (2002). Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, 7(1):13–22.
- Milliron, T., Jensen, R. J., Barzel, R., and Finkelstein, A. (2002). A framework for geometric warps and deformations. *ACM Transactions on Graphics*, 21(1):20–51.
- Möbius, A. F. (1827). *Der barycentrische Calcul*. Johann Ambrosius Barth, Leipzig.
- Nürnberg, G. and Zeilfelder, F. (2000). Developments in bivariate spline interpolation. *Journal of Computational and Applied Mathematics*, 121(1-2):125–152.
- Pinkall, U. and Polthier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36.
- Ruprecht, D. and Müller, H. (1995). Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15(2):37–43.
- Shewchuk, J. R. (2002). Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1-3):21–74.
- Sibson, R. (1980). A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87:151–155.
- Sibson, R. (1981). A brief description of natural neighbour interpolation. In Barnett, V., editor, *Interpolating Multivariate Data*, pages 21–36. Wiley, New York.
- Wachspress, E. L. (1975). *A Rational Finite Element Basis*. Academic Press, New York.
- Warren, J. (1996). Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics*, 6(2):97–108.
- Warren, J. (2003). On the uniqueness of barycentric coordinates. In *Contemporary Mathematics, Proceedings of AGGM '02*, pages 93–99.
- Warren, J., Schaefer, S., Hirani, A. N., and Desbrun, M. (2003). Barycentric coordinates for convex sets. Technical report, Rice University.
- Wolberg, G. (1990). *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos.

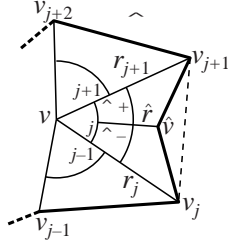


Figure 13: Notation for refining a polygon by adding a vertex.

A Refinability

If we refine the polygon Ψ to $\hat{\Psi}$ by adding a vertex \hat{v} between v_j and v_{j+1} as in Figure 13, then it follows from the locality of the homogeneous coordinates that $\hat{w}_i = w_i$ for $i \neq j, j+1$. According to Equation (11), the other three homogeneous coordinates are

$$\begin{aligned}\hat{w}_j(v) &= 2(\tan(\alpha_{j-1}(v)/2) + \tan(\hat{\alpha}^-(v)/2))/r_j(v), \\ \hat{w}(v) &= 2(\tan(\hat{\alpha}^-(v)/2) + \tan(\hat{\alpha}^+(v)/2))/\hat{r}(v), \\ \hat{w}_{j+1}(v) &= 2(\tan(\hat{\alpha}^+(v)/2) + \tan(\alpha_{j+1}(v)/2))/r_{j+1}(v).\end{aligned}$$

If we now write \hat{v} as an affine combination of v , v_j , and v_{j+1} , i.e.,

$$\hat{v} = \rho(v)v + \sigma(v)v_j + \tau(v)v_{j+1}$$

with barycentric coordinates

$$\sigma(v) = \frac{\hat{r}(v) \sin \hat{\alpha}^+(v)}{r_j(v) \sin \alpha_j(v)}, \quad \tau(v) = \frac{\hat{r}(v) \sin \hat{\alpha}^-(v)}{r_{j+1}(v) \sin \alpha_{j+1}(v)},$$

and

$$\rho(v) = 1 - \sigma(v) - \tau(v),$$

then some elementary transformations show that the homogeneous coordinates w_j and w_{j+1} can be expressed as

$$w_j(v) = \hat{w}_j(v) + \sigma(v)\hat{w}(v), \quad w_{j+1}(v) = \hat{w}_{j+1}(v) + \tau(v)\hat{w}(v),$$

and that the sum of the refined homogeneous coordinates satisfies

$$\hat{W}(v) = W(v) + \rho(v)\hat{w}(v).$$

It further follows that $\hat{W}(v) = W(v)$ in the special case that \hat{v} lies on the edge $e_j = (v_j, v_{j+1})$ since then $\rho(v) = 0$.

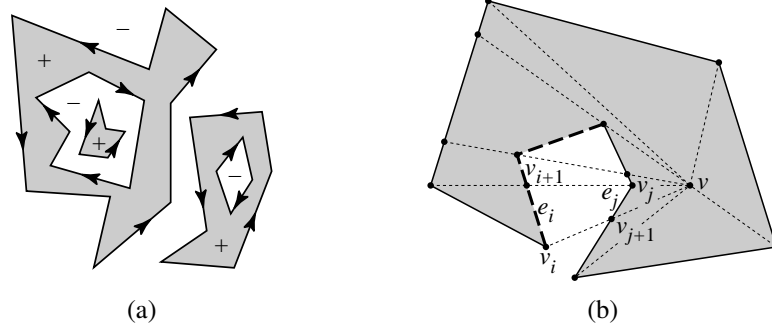


Figure 14: Interior and orientation of a set of polygons (a) and partitioning into sectors (b) with exit (solid) and entry edges (dashed).

B Normalization

The most important property of our homogeneous coordinates is that their sum $W(v)$ is never zero for any $v \notin \Psi$ and therefore the normalization in (3) is well-defined. We can even show that $W(v)$ is positive in the interior of Ψ and negative otherwise. For a set of polygons like in Figure 14 (a) we define the grey region as the interior and further assume the polygons to be orientated as indicated.

Now, let v be a vertex inside Ψ . We first consider the rays from v through the vertices v_i and add all intersection points to Ψ as shown in Figure 14 (b). According to Appendix A this does not change $W(v)$. Next we consider for any of the radial sectors between two neighbouring rays all edges inside the sector and classify them as *exit edges* (solid) or *entry edges* (dashed), depending on whether a traveller coming from v exits or enters the interior of Ψ at the edge. If we then define for each edge $e_i = (v_i, v_{i+1})$ the value

$$\kappa_i = \left(\frac{1}{r_i} + \frac{1}{r_{i+1}} \right) \tan(\alpha_i/2).$$

then it follows from the orientation of Ψ that κ_i is positive if e_i is an exit edge and negative if e_i is an entry edge. And for edges e_i that lie *on* one of the rays, we have $\kappa_i = 0$.

Now let e_i be one of the entry edges. Then there always exists an exit edge e_j in the same sector that is closer to v ; see Figure 14 (b). But as $\alpha_i = -\alpha_j$ and at least one of the inequalities $r_j \leq r_{i+1}$ and $r_{j+1} \leq r_i$ is strict, we have

$$\kappa_j = \left(\frac{1}{r_j} + \frac{1}{r_{j+1}} \right) \tan(\alpha_j/2) > \left(\frac{1}{r_i} + \frac{1}{r_{i+1}} \right) \tan(-\alpha_i/2) = -\kappa_i.$$

This means that the negative κ_i of the entry edge e_i is counterbalanced by the positive κ_j of the exit edge e_j . As this holds for all entry edges (without using any exit edge

twice) and also in the case that Ψ is a set of (possibly nested) polygons, we conclude that

$$\sum_{i=1}^n \kappa_i > 0.$$

But the sum of the κ_i can be rearranged, by a change of summation index, to be half the sum of the $w_i(v)$ in the form of Equation (11) and therefore $W(v)$ is positive for any v inside Ψ . Likewise one can show that $W(v)$ is negative over the exterior of Ψ .

C Lagrange and Linearity Property

From the representation of the homogeneous coordinates in Equation (11) it follows that $w_i(v)$ is C^∞ everywhere except at the vertices v_{i-1} , v_i , v_{i+1} and the edges e_{i-1} , e_i . As a consequence, the normalized coordinates $\lambda_i(v)$ are C^∞ at all $v \notin \Psi$. To study the behaviour of λ_i on one of the edges e_j of Ψ , we consider the slightly modified homogeneous coordinates

$$\hat{w}_i(v) = w_i(v)A_j(v) = \frac{r_j(v)r_{j+1}(v)}{r_i(v)}(z_{i-1}(v) + z_i(v)),$$

where $z_i(v) = \tan(\alpha_i(v)/2) \sin(\alpha_j(v))/2$. Then we have for any $v \in e_j$ that $z_i(v) = 0$ for $i \neq j$ and $z_j(v) = 1$ and therefore $\hat{w}_j(v) = r_{j+1}(v)$, $\hat{w}_{j+1}(v) = r_j(v)$, and $\hat{w}_i(v) = 0$ for $i \neq j, j+1$. It follows that $\lambda_i(v) = 0$ for all $i \neq j, j+1$ and

$$\lambda_j(v) = \frac{r_{j+1}(v)}{r_j(v) + r_{j+1}(v)}, \quad \lambda_{j+1}(v) = \frac{r_j(v)}{r_j(v) + r_{j+1}(v)},$$

which are the standard barycentric coordinates on an edge, converging to δ_{ij} as v approaches v_j . Further it is clear that the functions \hat{w}_i are C^∞ over e_j and so are the λ_i .

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany

Editor of the series: Jürgen Dix

Technical editor: Wojciech Jamroga

Contact: wjamroga@in.tu-clausthal.de

URL: <http://www.in.tu-clausthal.de/~wjamroga/techreports/>

The IfI Review Board

Prof. Dr. Jürgen Dix (Theoretical Computer Science/Computational Intelligence)

Prof. Dr. Klaus Ecker (Applied Computer Science)

Prof. Dr. habil. Torsten Grust (Databases)

Prof. Dr. Barbara Hammer (Theoretical Foundations of Computer Science)

Prof. Dr. Kai Hormann (Computer Graphics)

Dr. Michaela Huhn (Economical Computer Science)

Prof. Dr. Gerhard R. Joubert (Practical Computer Science)

Prof. Dr. Ingbert Kupka (Theoretical Computer Science)

Prof. Dr. Wilfried Lex (Mathematical Foundations of Computer Science)

Prof. Dr. Jörg Müller (Agent Systems)

Prof. Dr.-Ing. Dr. rer. nat. habil. Harald Richter (Technical Computer Science)

Prof. Dr. Gabriel Zachmann (Virtual Reality)